中国科学 : 信息科学
SCIENTIA SINICA Informationis
人工智能应用中的数据安全专刊 · 论文
《中国科学》杂志社
SCIENCE CHINA PRESS

SDFL: 一种隐私保护和拜占庭鲁棒的去中心化联邦学习方案

全韩彧[1,2], 钱彦屹[1,2], 田晖[1,2*], 刘雪峰[3], 李越[1,2], 王健宗[4], 钟迦[5]

1. 华侨大学计算机科学与技术学院, 厦门 361021
2. 厦门市数据安全与区块链技术重点实验室, 厦门 361021
3. 西安电子科技大学网络与信息安全学院, 西安 710071
4. 平安科技 (深圳) 有限公司, 深圳 518046
5. 福建泉工股份有限公司, 泉州 362123
* 通信作者. E-mail: htian@hqu.edu.cn

收稿日期: 2025–05–26; 修回日期: 2025–07–22; 接受日期: 2025–09–25; 网络出版日期: 2025–10–29

中央高校基本科研业务费专项 (批准号: ZQN-1207) 和国家重点研发计划课题 (批准号: 2023YFC3304501) 资助项目

摘要 去中心化联邦学习不依赖中央服务器, 解决了传统联邦学习中存在的服务器单点故障问题, 但同时也带来了更严重的隐私泄露和拜占庭攻击风险. 现有的隐私保护与拜占庭鲁棒研究主要面向传统联邦学习, 难以适用于去中心化场景. 为此, 本文提出了一种隐私保护和拜占庭鲁棒的去中心化联邦学习方案 SDFL (secure decentralized federated learning). 该方案采用客户端 – 委员会架构, 通过随机选择的委员会进行模型聚合, 基于可验证秘密共享和零知识证明技术设计隐私保护与抗拜占庭攻击方法. 安全性分析表明 SDFL 在去中心化场景下兼具隐私性与拜占庭鲁棒性; 采用多种机器学习模型在 MNIST 和 Fashion-MNIST 数据集上的实验结果进一步验证了 SDFL 能够有效抵抗拜占庭攻击, 同时保障模型分类准确率与高效性.

关键词 联邦学习, 隐私保护, 拜占庭鲁棒, 可验证秘密共享, 零知识证明

1 引言

随着人工智能技术和产业的发展, 数据分布呈现出高度分散的趋势, 数据孤岛问题普遍存在, 传统的集中式机器学习难以满足现代智能系统对数据安全和隐私的要求. 联邦学习 (federated learning, FL) 作为一种分布式机器学习范式 [1], 因其数据不离开本地的特点, 在金融、医疗等数据高度敏感行业受到广泛关注. 在标准的联邦学习框架中, 多个客户端在中央服务器的协调下训练本地模型, 由中央服务器对客户端的本地更新进行聚合得到全局模型.

然而, 在隐私保护和安全方面, 联邦学习仍然面临一系列问题与挑战 [2,3]. 首先, 已有研究表明, 联邦学习客户端的本地更新会泄露数据隐私, 容易受到成员推理攻击 [4]、数据重构攻击 [5,6] 等隐私攻击

引用格式: 全韩彧, 钱彦屹, 田晖, 等. SDFL: 一种隐私保护和拜占庭鲁棒的去中心化联邦学习方案. 中国科学: 信息科学, 2025, doi: 10.1360/SSI-2025-0208
Quan H Y, Qian Y Y, Tian H, et al. SDFL: a privacy-preserving Byzantine-robust decentralized federated learning scheme. Sci Sin Inform, 2025, doi: 10.1360/SSI-2025-0208

的威胁. 其次, 由于其分布式架构, 联邦学习还面临拜占庭攻击的风险. 恶意客户端可能上传伪造的本地更新, 干扰全局模型聚合, 降低模型的分类准确率 (投毒攻击) [7], 或诱导模型在特定任务上输出错误的结果 (后门攻击) [8].

除了隐私与安全问题, 传统的联邦学习依赖中央服务器执行关键的聚合操作, 还存在系统可靠问题. 例如, 中央服务器可能出现单点故障导致联邦学习系统无法工作, 或因其计算资源受限成为系统瓶颈, 甚至可能成为网络攻击的脆弱目标. 为了提升联邦学习系统的可靠度, 研究者们提出了去中心化联邦学习 (decentralized federated learning, DFL) [9]. 在 DFL 中, 模型的聚合过程由多个客户端合作完成, 取消了对中央服务器的依赖. 由于聚合任务不再集中于单点, 系统具备更好的容错能力.

但是, 去中心化架构进一步加剧了联邦学习的隐私和安全威胁. 在隐私威胁方面, DFL 客户端需要把本地更新分发给多个其他客户端, 增加了隐私泄露的风险. 在拜占庭攻击方面, 传统的联邦学习通常假设中央服务器是可信的 (或半可信的), 可由中央服务器执行拜占庭攻击检测. 然而, 在去中心化联邦学习中, 任何客户端都有可能是恶意的, 不存在单一可信的节点执行检测操作. 更严重的是, 在去中心化联邦学习中, 客户端不仅参与本地训练, 还参与模型聚合, 恶意客户端可能在不同的阶段发起攻击, 进一步增大了安全防御的难度.

针对以上问题, 研究者们开展了一系列联邦学习隐私保护与抗拜占庭攻击研究工作. 例如, 针对隐私攻击的安全聚合方法 [10,11], 以及针对拜占庭攻击的防御方法 [12,13]. 在传统的联邦学习场景下, 研究者们还提出了一系列同时解决隐私保护和拜占庭鲁棒问题的联邦学习方案 [14~20]. 但是, 这些方案都依赖于中央服务器执行安全聚合及拜占庭检测操作, 难以应用于无中央服务器的去中心化架构. 目前, 针对去中心化联邦学习, 只有少量的工作提出了能同时解决隐私保护和拜占庭鲁棒问题的方案 [21,22]. 然而, 这些工作都是基于差分隐私机制, 其添加的噪声在一定程度上会影响模型的准确率.

综上, 本文试图解决: 如何在保持模型分类准确率的前提下, 设计兼具隐私保护和拜占庭鲁棒功能的去中心化联邦学习方案?

针对该问题, 本文采用去中心化联邦学习中常见的客户端 – 委员会架构 [23,24], 由随机选择的委员会替代传统联邦学习中的中央服务器. 在该架构中, 客户端和委员会中均可能存在恶意节点. 为了抵抗拜占庭攻击, 本文提出了零知识证明技术结合共识机制的方法, 通过 “验证 – 共识” 实现恶意客户端和恶意委员会成员的检测, 再由合法委员会成员通过安全计算的方式实现隐私保护的模型聚合. 本文的主要贡献如下.

(1) 提出了一种新颖的去中心化联邦学习方案 SDFL (secure decentralized federated learning). 相比于基于差分隐私的方案, SDFL 不会造成联邦学习模型准确率下降.

(2) 在 SDFL 中, 基于 Feldman 可验证秘密共享 (Feldman’s verifiable secret sharing, FVSS) 与零知识证明, 创新地构造了一种 FVSS- 拜占庭检测联合电路, 能够在隐私保护的前提下高效地检测恶意客户端.

(3) 在 MNIST 与 Fashion-MNIST 两个公开数据集上, 以逻辑回归、多层感知机与卷积神经网络为测试模型, 系统评估了 SDFL 在拜占庭攻击下的分类准确率、计算与通信开销等关键指标, 实验结果表明 SDFL 具有良好的拜占庭鲁棒性和高效性.

本文剩余部分组织如下: 第 2 节和第 3 节分别对相关工作和预备知识进行介绍, 第 4 节描述系统与威胁模型及设计目标, 第 5 节对 SDFL 的方案细节进行详细描述, 第 6 和 7 节分别是安全性分析和实验分析, 最后对本文进行总结.

2　相关工作

自从联邦学习被提出以来, 其隐私保护和拜占庭鲁棒问题一直备受关注. 部分研究工作仅关注其中的一个问题. 例如, 针对隐私保护问题, 一种主流的方法是采用同态加密技术, 对本地更新进行加密

表 1　SDFL 与相关工作技术对比. ◐ 表示模型准确率会受到差分隐私噪声的影响.

Table 1 The technical comparison of SDFL with related studies. ◐ indicates that the model's accuracy is affected by differential privacy noise.

Scheme	System model	Privacy preservation	Byzantine robustness	Accuracy
[14]	Two non-colluding servers	Crypto	Hamming distance	●
[15]	Two non-colluding servers	Crypto	K-means	●
[16]	Single server	Crypto	Multi-Krum	●
[17]	Single server	Crypto	L_2 norm bound	●
[18]	Single server	Crypto	Generic per-client check	●
[19]	Single server	Differential privacy	LFH	◐
[20]	Single Server	Differential privacy	K-means	◐
[21]	Decentralized	Differential privacy	Generic	◐
[22]	Decentralized	Differential privacy	Jaccard similarity	◐
SDFL	Decentralized	Crypto	L_2 norm bound	●

后聚合 [11]; 针对拜占庭鲁棒问题, Guo 等 [25] 提出了联邦学习用户信任评估方法, Yang 等 [26] 则在最新的工作中提出了综合余弦相似度、符号统计和谱方法的多策略防御方法. 近年来, 研究者们还提出了一系列同时解决这两个问题的方案. 表 1 列出了本文的 SDFL 方案与主要相关工作的技术对比. 从隐私保护的角度, 这些相关工作可以分为基于密码技术 [27] 和基于差分隐私技术 [28] 两大类.

基于密码技术: Miao 等 [14] 提出了一个基于双服务器架构的安全联邦学习方案, 利用汉明距离 (Hamming distance) 检测恶意客户端, 通过双服务器之间的安全计算实现隐私保护. 穆旭彤等 [15] 也采用双服务器架构, 提出了一个基于安全多方计算的抗拜占庭攻击的隐私保护联邦学习方案, 通过安全聚类实现恶意客户端检测, 同时采用随机 PCA 降维技术提高计算性能. So 等 [16] 在单服务器架构下提出了一个抗拜占庭攻击的安全聚合方案 BREA. BREA 采用可验证秘密共享技术, 在用户之间安全共享联邦学习本地更新, 在服务器的辅助下利用 Multi-Krum 算法实现恶意客户端检测. 同样在单服务器架构下, Lycklama 等 [17] 采用零知识证明技术, 提出了一个基于 BulletProofs 零知识证明系统 [29] 的抗拜占庭攻击安全联邦学习方案 RoFL. BulletProofs 可以对取值范围进行高效证明, 因此 RoFL 采用 L_2 范数边界作为拜占庭攻击检测方法, 结合基于掩码的安全聚合方法 [30] 保护隐私. Roy 等 [18] 对可验证输入的安全聚合 (secure aggregation of verified inputs, SAVI) 进行了形式化描述, 基于 SNIP 零知识证明系统 [31] 提出了一个通用的安全联邦学习框架 EIFFeL. 相比于 RoFL, EIFFeL 支持所有通用的单客户端拜占庭检测方法 (例如 L_2 范数边界、Zeno++ [32]、余弦相似度等).

基于差分隐私: 差分隐私是联邦学习隐私保护的重要技术手段. 在隐私保护的拜占庭鲁棒联邦学习方案设计方面, 文献 [33] 提出了一个基于分布式差分隐私和范围证明的联邦学习方案, 采用分布式差分隐私实现隐私保护, 利用 BulletProofs 证明本地更新参数在限定的范围内, 从而实现隐私保护和拜占庭鲁棒的联邦学习. Zhu 等 [34] 在工作 [35] 提出的拜占庭鲁棒随机聚合方法 RSA 的基础上引入差分隐私, 提出了 DP-RSA 方法, 通过对本地模型和全局模型差值的正负 (sign) 进行扰动实现隐私保护的拜占庭鲁棒聚合. Gu 等 [19] 基于 LFH 拜占庭鲁棒优化方法 [36], 提出了一种差分隐私和拜占庭鲁棒的联邦学习方案 DP-BREM, 相比于文献 [33, 34] 的方案具有更好的分类准确率. 另一方面, 文献 [37, 38] 的研究工作表明差分隐私噪声自身能在一定程度上提高联邦学习模型鲁棒性. 然而, Qi 等 [20] 通过实验证明了差分隐私只能抵抗有限的攻击手段, 同时提出了一个基于瑞丽 (Rényi) 差分隐私的联邦学习方案 Robust-DPFL. 相比于基于密码学的方案, 基于差分隐私的方案通常更加高效, 但是差分隐私噪声在一定程度上会影响联邦学习模型的准确率.

以上研究工作都是面向传统的中心化联邦学习系统, 依赖中央服务器 (或两个服务器) 实现恶意

客户端检测, 因此无法应用于去中心化联邦学习系统. 目前, 在去中心化联邦学习架构下, 同时解决隐私保护和拜占庭鲁棒问题的研究工作较少. 面向去中心化场景, Ye 等 [21] 提出了首个隐私保护和拜占庭鲁棒的去中心化随机梯度下降框架. Zhang 等 [22] 提出了首个隐私增强和抗拜占庭攻击的个性化联邦学习方法. 但是, 他们的方案都是基于差分隐私技术, 模型分类准确率会随着隐私保护能力的增强 (即隐私预算的减小) 而降低. 相比之下, 本文提出的 SDFL 利用可验证秘密共享实现隐私保护, 不会损失模型的分类准确率.

3　预备知识

3.1　Feldman 可验证秘密共享

Feldman 可验证秘密共享 (FVSS) [39] 是一种门限秘密共享方案. 在一个 (m,k)-FVSS 方案中, k 个参与者共享一个秘密 $s\in\mathbb{Z}_p$, 每个参与者持有 s 的一个份额, 使得 (1) 任何 m 个参与者都能计算出 s 的值; (2) 任何 $m-1$ 或更少的参与者都不能得到 s 的任何信息; (3) 给定一个份额, 可以验证该份额是否合法. (m,k)-FVSS 由以下 3 个算法组成.

- $\{\langle s\rangle,\Psi\}\leftarrow FVSS.Share(s,m,k)$: 给定秘密 $s\in\mathbb{Z}_p$, 随机选择 $a_1,\cdots,a_{m-1}\in\mathbb{Z}_p$, 令 $a_0=s$, 构造多项式如下:

$$f(x)=a_0+a_1x+\cdots+a_{m-1}x^{m-1}\bmod p, \tag{1}$$

计算 $\langle s\rangle_i=f(i)$, 输出 s 的 k 个份额 $\langle s\rangle=\{\langle s\rangle_1,\langle s\rangle_2,\cdots,\langle s\rangle_k\}$, 以及用于验证份额合法性的承诺 Ψ.

- $b\leftarrow FVSS.Verify(\langle s\rangle_j,\Psi)$: 给定秘密 s 的一个份额 $\langle s\rangle_j$ 及承诺 Ψ, 验证 $\langle s\rangle_j$ 是否合法 (即 $\langle s\rangle_j\stackrel{?}{=}f(j)$), 若通过验证, 则输出 1, 否则输出 0.

- $s\leftarrow FVSS.Recon(\{\langle s\rangle_{i_j}\}_{j=1}^m)$: 给定任意 m 个份额, 利用拉格朗日 (Lagrange) 插值公式计算出 s 的值.

此外, FVSS 是线性的. 设 $\{\langle s_1\rangle_i\}_{i=1}^k$ 和 $\{\langle s_2\rangle_i\}_{i=1}^k$ 分别是秘密 s_1 和 s_2 的一组份额, 则 $\{\langle s_1\rangle_i+\langle s_2\rangle_i\}_{i=1}^k$ 是 s_1+s_2 的一组合法份额, 即由任意 m 个份额 $\{\langle s_1\rangle_{i_j}+\langle s_2\rangle_{i_j}\}_{j=1}^m$ 可以计算出 s_1+s_2 的值. FVSS 的共享份额是信息论安全的, 而承诺的安全性依赖于离散对数困难问题假设.

3.2　零知识证明

零知识证明 (zero-knowledge proof, ZKP) 是由 Goldwasser 等 [40] 最早提出的. 在 ZKP 中, 证明者能向验证者证明某个陈述是正确的, 且不泄露其他信息. ZKP 具有 3 个性质: (1) 完备性, 即给定某个陈述及其证据, 证明者能使验证者相信该陈述是正确的; (2) 可靠性, 即恶意的证明者无法使验证者相信一个错误的陈述; (3) 零知识, 即除了陈述的正确性, 证明不泄露其他任何信息, 特别是证明者的证据. 本文采用简洁非交互零知识证明 (zero-knowledge succinct non-interactive argument of knowledge, zk-SNARK) 技术 [41], 即证明者只要向验证者发送一轮消息即可完成证明.

给定有限域 $\mathbb{F}$ (例如 $\mathbb{Z}_p$), 设陈述为 $x\in\mathbb{F}^n$, 证据为 $w\in\mathbb{F}^h$, 陈述和证据的关系 $\mathcal{R}_C$ 用 $\mathbb{F}$- 算术电路 $C(x;w)=0^l$ 描述, 其中 x 是电路的公开输入, w 是秘密输入. 对于给定的电路 C, zk-SNARK 由以下 3 个算法组成.

- $(PK,VK)\leftarrow ZK.KeyGen(1^\lambda,C)$: 给定安全参数 λ, 输出 C 的证明密钥 PK 和验证密钥 VK.

- $\pi\leftarrow ZK.Prove(PK,x,w)$: 给定证明密钥 PK、陈述 x 和证据 w, 输出证明 π.

- $b\leftarrow ZK.Verify(VK,x,\pi)$: 给定验证密钥 VK、陈述 x 和证明 π, 如果证据 π 是合法的, 则输出 $b=1$, 否则输出 $b=0$.

通过 zk-SNARK, 证明者向验证者证明 x 和 w 满足关系 $\mathcal{R}_C$, 即 $C(x,w)=0^l$ 成立, 且不透露证据 w 的信息.

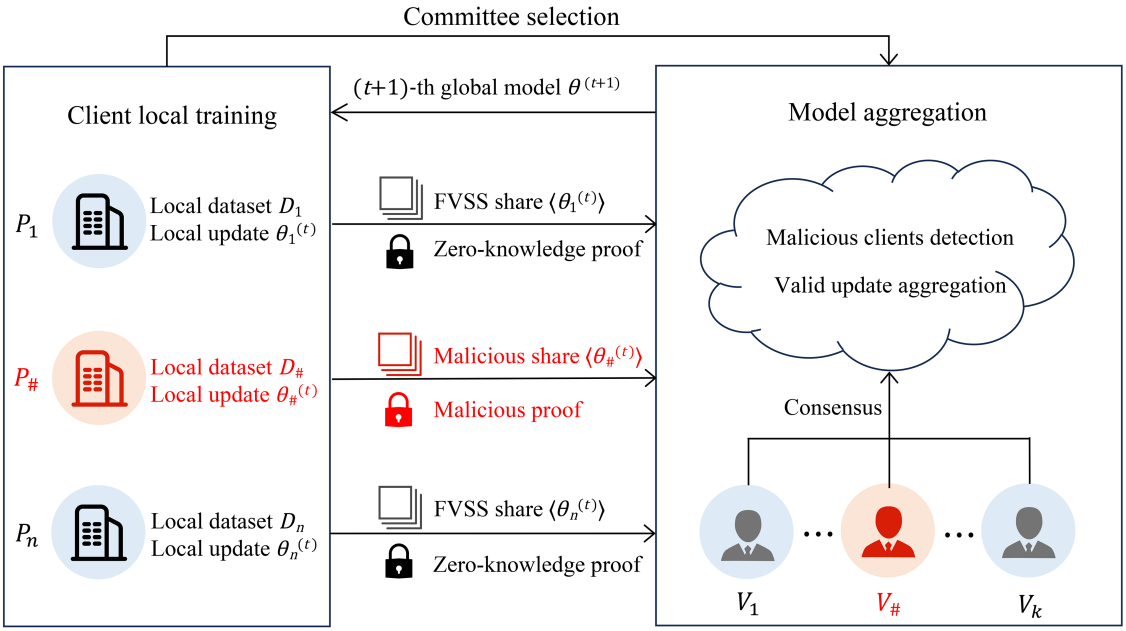

图 1　(网络版彩图) SDFL 系统模型.

Figure 1　(Color online) System model of SDFL.

4　问题描述

本节详细介绍 SDFL 的系统模型、威胁模型及设计目标.

4.1　系统模型

如图 1 所示, SDFL 系统主要由客户端和委员会组成. 设系统中共有 n 个客户端, 记为 $P = \{P_1, \cdots, P_n\}$, 每个客户端拥有一个本地数据集 $\mathcal{D}_i$, 其中 $i \in [n]$. 委员会成员从客户端中随机选取. 假设委员会中共有 k 个成员, 记为 $V = \{V_1, \cdots, V_k\} \subset P$. 在联邦学习的第 t 轮, 各实体执行如下操作.

- 每个客户端 P_i 获取第 t 轮的全局模型 $\theta^{(t)}$ 后, 使用 $\mathcal{D}_i$ 对 $\theta^{(t)}$ 进行本地更新, 得到 $\theta_i^{(t)}$, 以秘密共享的方式发送给委员会, 同时采用 zk-SNARK 向委员会证明 $\theta_i^{(t)}$ 是合法的.
- 每个委员会成员 V_j 收到客户端发来的本地更新 $\theta_i^{(t)}$ 后, 首先检测其是否为恶意更新 (即拜占庭更新), 得到合法客户端列表 P^*, 然后通过计算 $\theta^{(t+1)} \leftarrow \sum_{P_i \in P^*} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \theta_i^{(t)}$ 对合法的本地更新进行聚合, 生成下一轮的全局模型 $\theta^{(t+1)}$. 委员会成员本身也以客户端身份参与联邦学习.

此外, 本文假设系统中存在一个可信第三方 TA (trusted authority) 和一个可公开访问的公告板 $\mathcal{B}$. TA 负责系统初始化, 包括系统公开参数设置和零知识证明密钥生成. $\mathcal{B}$ 负责记录客户端广播发布的消息. 可信第三方和公告板是联邦学习系统中常见的实体 [18, 42].

4.2　威胁模型

SDFL 系统主要面对隐私攻击和拜占庭攻击两类威胁.

- 隐私攻击: 假设所有客户端 (包括委员会成员) 都是诚实但好奇的, 即他们会正确地执行规定的操作, 但是企图从接收到的消息中推测出客户端本地数据的隐私信息.
- 拜占庭攻击: 假设客户端和委员会成员中均可能存在恶意节点, 他们以破坏联邦学习模型分类准确率为目标, 可能不按照协议规定执行任意操作. 例如, 恶意客户端可能上传错误的本地更新参数, 或提供虚假的本地更新零知识证明; 而恶意委员会成员可能故意对恶意客户端提供的本地更新进行聚合, 或故意在聚合过程中排除合法的本地更新等.

在 SDFL 中, 假设恶意攻击者的数量最多为 $m-1$ 个, 且客户端、委员会成员、恶意攻击者的数量关系满足 $n \geqslant k \geqslant 2m$, 即委员会中诚实者 (诚实但好奇的) 占大多数.

4.3　设计目标

SDFL 的设计目标如下.

• 隐私性: SDFL 的首要目标是保护客户端的本地数据隐私. 由于本地更新的模型参数可能泄露数据, SDFL 应确保系统中的任何诚实但好奇的实体无法获取客户端的本地更新.

• 拜占庭鲁棒性: 针对可能存在的恶意客户端和恶意委员会成员, SDFL 应确保委员会只对合法的本地更新进行聚合, 从而避免恶意更新对模型分类准确率的影响.

• 高效性: 在保证隐私保护和抗拜占庭攻击的前提下, SDFL 还应优化计算和通信效率, 确保系统具备较低的开销.

5　SDFL 方案描述

5.1　设计思想

SDFL 要解决的核心问题是去中心化架构下的联邦学习隐私保护与拜占庭鲁棒问题. SDFL 采用如下的设计思想: 首先, 为了解决本地更新隐私保护问题, 客户端通过 FVSS 在委员会之间安全地共享本地更新, 利用 FVSS 实现隐私保护的本地更新聚合. 理论上, 委员会成员也可以采用安全多方计算技术实现拜占庭攻击的检测. 但是, 由于拜占庭攻击检测方法通常涉及比较大小等非线性运算 (例如 Krum、范数边界等), 在采用安全多方计算时会带来较高的计算开销, 且通信开销会随着委员会成员数量的增加而显著上升, 尤其在委员会中可能存在恶意节点的情况下. 为此, SDFL 利用简洁非交互零知识证明 (zk-SNARK), 由客户端向委员会主动证明其本地更新是合法的, 从而提高系统性能. 但是, 该方法需要解决两个技术挑战: 第一, 针对恶意客户端, 如何确保用于 FVSS 和 zk-SNAKR 的本地更新是相同的; 第二, 针对恶意委员会成员, 如何确保合法本地更新的正确聚合.

针对第一个挑战, 本文构造了如图 2 所示的 FVSS- 拜占庭检测联合电路, 将 FVSS 计算过程与拜占庭检测过程绑定, 通过共享本地更新秘密输入, 确保用于 FVSS 和拜占庭检测的本地更新是相同的; 针对第二个挑战, 本文构建了一个由委员会成员共同维护的信任矩阵 (详见 5.3.4 节), 通过共识机制识别恶意委员会成员.

在具体描述 SDFL 的方案细节之前, 首先介绍 SDFL 中的核心零知识电路设计.

5.2　零知识证明电路

给定陈述 x 和证据 w, 使用 zk-SNARK 的关键是把待证明的知识 (即 x 和 w 的关系 $\mathcal{R}_C$) 表示成电路的形式. 本节详细介绍 SDFL 中涉及的零知识证明电路. 为了便于描述, 对于电路 $C(x;w)$, 默认分号前的参数为电路的公开输入, 分号后的参数为电路的秘密输入. 若 x 和 w 满足 $\mathcal{R}_C$, 则 $C(x;w)$ 输出 0.

5.2.1　拜占庭检测电路

本文采用联邦学习中常用的范数边界方法 (norm bound) [43] 作为拜占庭检测算法. 范数边界方法具有简洁高效的特点, 特别适用于基于零知识证明、安全计算等密码技术的隐私保护系统 [17,18,44]. 具体地, 给定本地更新 $\theta_i^{(t)} \in \mathbb{F}^d$ 和公开阈值 τ, 若 $\theta_i^{(t)}$ 的 L_2 范数 $||\theta_i^{(t)}||_2 < \tau$, 则判定 $\theta_i^{(t)}$ 为合法的本地更新, 否则判定 $\theta_i^{(t)}$ 为恶意本地更新. τ 可由公开数据集计算得到.

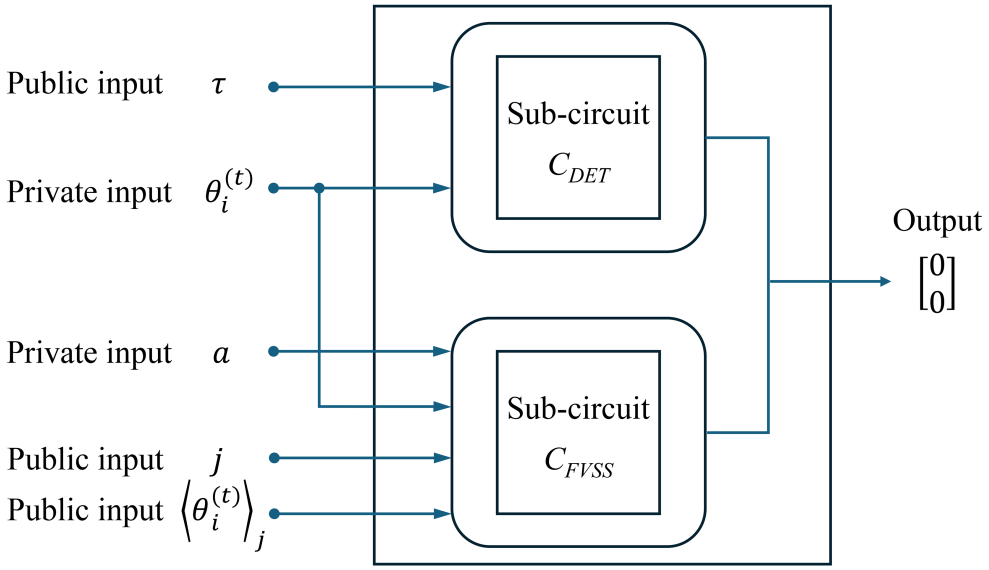

图 2　(网络版彩图) FVSS- 拜占庭检测联合电路 C_{COMB}.

Figure 2　(Color online) Combined circuit of FVSS and Byzantine detection C_{COMB}.

在 SDFL 中, 客户端 P_i 在本地利用 zk-SNARK 证明其本地更新 $\theta_i^{(t)}$ 的范数小于给定的阈值 τ, 若委员会验证通过, 则证明 $\theta_i^{(t)}$ 是合法的, 具体电路为

$$C_{DET}(\tau;\theta_i^{(t)}) := C_{COMP}(||\theta_i^{(t)}||_2;\tau), \tag{2}$$

其中, C_{COMP} 是比较大小电路, 若 $||\theta_i^{(t)}||_2 < \tau$, 则输出 0, 反之输出 1. 由于比较大小在 zk-SNARK 中已有成熟的实现方法 [45], 本文不对其细节展开描述.

5.2.2　FVSS- 拜占庭检测联合电路

给定客户端 P_i 本地更新 $\theta_i^{(t)}$ 的一组共享 $\langle\theta_i^{(t)}\rangle = \{\langle\theta_i^{(t)}\rangle_j\}_{j=1}^k$, 每个委员会成员 V_j 持有其中的一个份额 $\langle\theta_i^{(t)}\rangle_j$. 根据 FVSS 的性质, V_j 可以执行 $FVSS.Verify$ 算法验证其持有的份额是严格按照 $FVSS.Share(\theta_i^{(t)},m,k)$ 计算出来的. 然而, 该验证操作无法证明客户端共享的是通过拜占庭检测的本地更新. 换句话说, 恶意客户端可能在拜占庭检测电路中输入合法的本地更新, 而共享的是恶意更新, 即在 $C_{DET}(\tau;\theta_i^{(t)})$ 和 $FVSS.Share(\theta_i^{(t)},t,k)$ 中输入不同的 $\theta_i^{(t)}$. 为了解决该问题, 本文构建了 FVSS- 拜占庭检测联合电路 C_{COMB} (图 2), 具体如下:

$$C_{COMB}(\tau,\langle\theta_i^{(t)}\rangle_j,j;\theta_i^{(t)},a) := \begin{bmatrix} C_{DET}(\tau;\theta_i^{(t)}) \\ C_{FVSS}(\langle\theta_i^{(t)}\rangle_j,j;\theta_i^{(t)},a) \end{bmatrix}, \tag{3}$$

其中, C_{FVSS} 是 FVSS 子电路, $a=\{a_1,\cdots,a_{m-1}\}$ 是 (m,k)-FVSS 中的多项式系数, 具体如下:

$$C_{FVSS}(\langle\theta_i^{(t)}\rangle_j,j;\theta_i^{(t)},a) := \langle\theta_i^{(t)}\rangle_j - (\theta_i^{(t)} + a_1\cdot j + \cdots + a_{m-1}\cdot j^{m-1}). \tag{4}$$

如图 2 所示, 在电路 C_{COMB} 中, 子电路 C_{DET} 和子电路 C_{FVSS} 共享秘密输入 $\theta_i^{(t)}$, 确保了通过拜占庭检测和 FVSS 共享的本地更新是相同的.

5.3　方案详细描述

协议 1 描述了 SDFL 的方案细节, 分为 4 个阶段: 系统初始化、委员会选举、本地训练和模型聚合, 其中模型聚合包括恶意客户端检测和本地更新聚合两个子阶段.

- **系统初始化**

  TA 设置安全参数 λ 和有限域 $\mathbb{Z}_p$, 执行 $(PK, VK) \leftarrow ZK.KeyGen(1^\lambda, C_{COMB})$, 设置安全散列函数 $H(\cdot)$, 发送 $(p, PK, VK, H(\cdot))$ 至公告板 $\mathcal{B}$.

- **委员会选举**

  所有客户端 $P_1, P_2, \cdots, P_n$ 共同执行算法 1, 选举出委员会成员 $V = \{V_1, V_2, \cdots, V_k\}$.

- **本地训练**

  每个客户端 P_i:

  – 从公告板 $\mathcal{B}$ 读取当前全局模型 $\theta^{(t)}$.

  – 执行算法 2, 得到本地更新 $\theta_i^{(t)}$ 的 k 个 FVSS 份额 $\{\langle\theta_i^{(t)}\rangle_j\}_{j=1}^k$, FVSS 承诺 $\psi_i^{(t)}$, 以及对应的 k 个零知识证明 $\{\pi_{ij}\}_{j=1}^k$.

  – 发送 $\Psi_i^{(t)}$ 至公告板 $\mathcal{B}$, 通过安全信道发送 $(\langle\theta_i^{(t)}\rangle_j, \pi_{ij})$ 给委员会成员 V_j.

- **模型聚合 (恶意客户端检测)**

  每个委员会成员 V_j:

  – 执行算法 3, 得到 n 维信任向量 δ_j.

  – 上传 δ_j 至公告板, 组成信任矩阵 $\Delta = [\delta_1, \delta_2, \cdots, \delta_k]^{\mathrm{T}}$.

  – 初始化合法客户端列表 $P^* = \emptyset$ 及合法委员会成员列表 $V^* = \{V_1, \cdots, V_k\}$, 采用如下共识机制更新 P^* 和 V^*:

  → 若 $\sum_{\ell=1}^k \delta_{\ell i} \geqslant m$ 成立, 则把 P_i 加入 P^*.

  → 若 $(\delta_{\ell i} == 0) \wedge (P_i \in P^*)$ 成立, 或 $(\delta_{\ell i} == 1) \wedge (P_i \notin P^*)$ 成立, 则把 V_ℓ 移出 V^*.

- **模型聚合 (本地更新聚合)**

  V^* 中的每个委员会成员 V_j 执行:

  – 计算全局模型的 FVSS 份额 $\langle\theta^{(t+1)}\rangle_j = \sum_{P_i \in P^*} \langle\theta_i^{(t)}\rangle_j$, 并上传到公告板 $\mathcal{B}$.

  – 对于 $1 \leqslant i \leqslant |V^*|$, 遍历公告板上所有 $\langle\theta^{(t+1)}\rangle_i$ 的 m 个份额的组合, 执行 $FVSS.Recon$ 得到所有可能的全局模型, 其中出现最多的计算结果即为下一轮全局模型 $\theta^{(t+1)}$, 把该结果发送至公告板进行下一轮本地训练.

协议 1　SDFL 的方案细节.

5.3.1　系统初始化阶段

在 SDFL 中, TA 负责初始化系统公开参数. 具体地, TA 设置安全参数 λ 和有限域 $\mathbb{Z}_p$, 执行 $ZK.KeyGen(1^\lambda, C_{COMB})$, 生成零知识证明密钥 PK 和 VK. 此外, TA 设置一个安全散列函数 $H(\cdot)$. 最后, TA 将系统公开参数 $(p, PK, VK, H(\cdot))$ 发送至公告板 $\mathcal{B}$, 公开给所有客户端.

5.3.2　委员会选举阶段

委员会成员从所有客户端中随机选择, 具体过程如算法 1 所示. 首先, 每个客户端 P_i 在本地生成一个随机数 $r_i \in \mathbb{Z}_p$, 发送 r_i 的散列值 $h_i = H(r_i)$ 至公告板 $\mathcal{B}$. 在所有客户端发送完散列值后, 每个客户端再发送 r_i 至公告板. 然后, 每个客户端读取公告板上的所有随机数, 计算公共随机数 $r = \sum_{i=1}^n r_i$. 在该阶段, 散列值 h_i 作为随机数 r_i 的承诺, 可防止恶意客户端根据其他合法客户端的随机数修改自己的随机数. 然后, 每个客户端 P_i 在本地根据通过公共随机数 r 计算出的 k 个委员会成员索引 $i_1, i_2, \cdots, i_k$, 选举出委员会 $V = \{V_1, V_2, \cdots, V_k\} = \{P_{i_1}, P_{i_2}, \cdots, P_{i_k}\}$. 因为公共随机数是公开且唯一的, 所以各客户端得到的委员会成员名单是相同的, 且由于该名单是各客户端在本地独立计算的, 恶意客户端无法冒充委员会成员.

算法 1 委员会选举 (committee selection).

Input: 客户端数 n, 委员会成员数 k.
Output: 委员会成员列表 $V = \{V_1, V_2, \cdots, V_k\}$.

for $i = 1$ to n **do**
  P_i 随机生成 $r_i \in \mathbb{Z}_p$;
  P_i 发送 $h_i = H(r_i)$ 至 $\mathcal{B}$;
end for
for $i = 1$ to n **do**
  P_i 发送 r_i 至 $\mathcal{B}$;
end for
for $i = 1$ to n **do**
  P_i 从 $\mathcal{B}$ 读取 $r_1, r_2, \cdots, r_n$;
  P_i 计算 $r = r_1 + r_2 + \cdots + r_n$;
  for $j = 1$ to k **do**
    P_i 计算 $i_j = (H(r||j) \bmod n) + 1$;
  end for
  P_i 输出 $V = \{V_1, V_2, \cdots, V_k\} = \{P_{i_1}, P_{i_2}, \cdots, P_{i_k}\}$;
end for

算法 2 本地训练 (local training).

Input: 全局模型 $\theta^{(t)}$, 本地数据集 $\mathcal{D}_i$.
Output: 本地更新 FVSS 份额 $\{\langle\theta_i^{(t)}\rangle_j\}_{j=1}^k$, FVSS 承诺 $\Psi_i^{(t)}$, 本地更新证明 $\{\pi_{ij}\}_{j=1}^k$.

P_i 更新 $\theta_i^{(t)} \leftarrow ClientUpdate(\theta^{(t)}, \mathcal{D}_i)$;
P_i 计算 $\{\langle\theta_i^{(t)}\rangle_1, \langle\theta_i^{(t)}\rangle_2, \cdots, \langle\theta_i^{(t)}\rangle_k, \Psi_i^{(t)}\} \leftarrow FVSS.Share(\theta_i^{(t)}, m, k)$;
for $j = 1$ to k **do**
  P_i 计算 $\pi_{ij} \leftarrow ZK.Prove(PK, (\tau, \langle\theta_i^{(t)}\rangle_j, j), (\theta_i^{(t)}, a_i))$;
  // τ 是 L_2 范数边界.
  // $a_i = \{a_{i1}, \cdots, a_{i(m-1)}\} \in \mathbb{Z}_p^{(m-1)}$ 是 $FVSS.Share(\theta_i^{(t)}, m, k)$ 中生成的多项式系数.
end for
return $\{\langle\theta_i^{(t)}\rangle_j\}_{j=1}^k$, $\psi_i^{(t)}$, $\{\pi_{ij}\}_{j=1}^k$.

5.3.3 本地训练阶段

在本地训练阶段, 客户端的主要操作如算法 2 所描述. 首先, 客户端 P_i 从公告板 $\mathcal{B}$ 获得当前全局模型 $\theta^{(t)}$, 利用 $\mathcal{D}_i$ 对 $\theta^{(t)}$ 进行训练, 得到本地更新 $\theta_i^{(t)}$ (第 1 行). 其中, $ClientUpdate$ 的具体计算过程由模型和优化方法确定, 例如采用随机梯度下降方法. 然后, P_i 执行 $FVSS.Share$ 生成本地更新的 k 个份额和用于验证份额的承诺 $\Psi_i^{(t)}$ (第 2 行). 接着, P_i 为每个份额 $\langle\theta_i^{(t)}\rangle_j$ 生成零知识证明 π_{ij} (第 4 行). 其中, $(\tau, \langle\theta_i^{(t)}\rangle_j, j)$ 是电路 C_{COMB} 的公开输入, $(\theta_i^{(t)}, a_i)$ 是秘密输入. 最后, 客户端 P_i 将 $(\langle\theta_i^{(t)}\rangle_j, \pi_{ij})$ 通过安全信道发送给委员会成员 V_j, 同时将 $\Psi_i^{(t)}$ 发布到公告板.

5.3.4 模型聚合阶段

SDFL 的模型聚合由委员会成员合作执行, 包括恶意客户端检测与合法本地更新聚合两个子阶段.

恶意客户端检测: 在本地训练阶段结束后, 每个委员会成员 V_j 持有 $\{\langle\theta_i^{(t)}\rangle_j, \pi_{ij}, \Psi_i^{(t)}\}_{i=1}^n$. 委员会首先检测是否存在恶意客户端. 具体地, 每个委员会成员 V_j 执行算法 3, 得到一个 n 维信任向量 δ_j. 其中, 针对每个客户端 P_i 的本地更新份额 $\langle\theta_i^{(t)}\rangle_j$, V_j 分别进行 FVSS 验证 (第 3 行) 和 zk-SNARK 验证 (第 4 行). 当且仅当两项验证都通过时, $\delta_{ji} = 1$, 否则 $\delta_{ji} = 0$. 完成验证后, V_j 将 δ_j 上传到公告板. 最后, 所有委员会成员上传的信任向量共同组成一个信任矩阵 $\Delta = [\delta_1, \delta_2, \cdots, \delta_k]^{\mathrm{T}}$.

算法 3 恶意客户端检测 (malicious clients detection).

Input: 本地更新 FVSS 份额 $\{\langle\theta_i^{(t)}\rangle_j\}_{i=1}^n$, FVSS 承诺 $\{\Psi_i^{(t)}\}_{i=1}^n$, 本地更新证明 $\{\pi_{ij}\}_{i=1}^n$.
Output: 信任向量 δ_j.

V_j 初始化 $\delta_j = [\delta_{j1}, \delta_{j2}, \cdots, \delta_{jn}] = [0, 0, \cdots, 0]$;
for $i = 1$ to n **do**
  V_j 计算 $b_{i1} \leftarrow FVSS.Verify((j, \langle\theta_i^{(t)}\rangle_j), \Psi_i^{(t)})$;
  V_j 计算 $b_{i2} \leftarrow ZK.Verify(VK, (\tau, \langle\theta_i^{(t)}\rangle_j, j), \pi_{ij})$;
  if $(b_{i1} \wedge b_{i2}) == 1$ **then**
    $\delta_{ji} = 1$ // P_i 是合法客户端;
  else
    $\delta_{ji} = 0$ // P_i 是恶意客户端;
  end if
end for
return δ_j;

Trust matrix Δ

	P_1	P_2	P_3	$\cdots$	P_{20}
V_1, δ_1	1	0	0	$\cdots$	1
V_2, δ_2	1	0	0	$\cdots$	1
V_3, δ_3	0	1	1	$\cdots$	1
V_4, δ_4	1	0	0	$\cdots$	1
V_5, δ_5	0	1	0	$\cdots$	1
V_6, δ_6	0	0	0	$\cdots$	1
V_7, δ_7	1	0	0	$\cdots$	1
V_8, δ_8	1	0	0	$\cdots$	1

图 3　(网络版彩图) 一个信任矩阵的例子 ($n = 20$, $k = 8$, $m = 4$).
Figure 3　(Color online) An example of trust matrix ($n = 20$, $k = 8$, $m = 4$).

为了防止恶意的委员会成员虚构信任向量 (即把合法的客户端记为恶意的, 或把恶意的客户端记为合法的), 委员会采用以下共识机制.

- 对于客户端 P_i, 若至少有 m 个委员会成员认为其是合法的, 即 $\sum_{\ell=1}^k \delta_{\ell i} \geqslant m$, 则该客户端为合法客户端, 否则该客户端为恶意客户端.
- 若 P_i 是通过共识的合法客户端, 但存在 $\delta_{\ell i} = 0$, 则 V_ℓ 是恶意委员会成员. 同理, 若 P_i 是通过共识的恶意客户端, 但存在 $\delta_{\ell i} = 1$, 则 V_ℓ 也是恶意委员会成员.

由于每个委员会成员在本地独立按照该共识机制检测恶意客户端, 最后, 各委员会成员得到一个共识的合法客户端列表 P^* 及一个共识的合法委员会成员列表 V^*.

如图 3 中的例子所示, 红色表示恶意节点, 根据共识机制, 虽然 3 个恶意委员会成员把 P_1 记为恶意客户端, 但是由于 $\sum_{j=1}^8 \delta_{j1} = 5 \geqslant 4$, P_1 仍被认为是合法客户端, 而 $\{V_3, V_5, V_6\}$ 则被判定为恶意委员会成员, 最后可得到 $P^* = \{P_1, P_4, P_5, \cdots, P_{20}\}$, $V^* = \{V_1, V_2, V_4, V_7, V_8\}$.

本地更新聚合: 在得到 P^* 和 V^* 后, V^* 中的委员会成员对合法本地更新 (即 P^* 中的客户端上

传的本地更新) 进行聚合, 计算出下一轮的全局模型. 具体地, V^* 中的每个委员会成员 V_j 计算聚合结果的 FVSS 份额 $\langle\theta^{(t+1)}\rangle_j = \sum_{P_i\in P^*}\langle\theta_i^{(t)}\rangle_j$, 并上传到公告板 $\mathcal{B}$. 最后, 公告板上共有 $|V^*|$ 个聚合结果的 FVSS 份额.

但是, 基于信任矩阵的共识机制只能保证 P^* 中的客户端是合法客户端, 而 V^* 中仍可能存在恶意委员会成员, 他们在恶意客户端检测阶段提供正确的信任向量, 因而未被发现. 这些恶意委员会成员可能上传了恶意的 FVSS 份额. 由于委员会成员的数量 $k \geqslant 2m$, 且至多有 $m-1$ 个恶意委员会成员, 所以至少有 $m+1$ 个合法委员会成员上传了合法的 FVSS 份额 $\langle\theta^{(t+1)}\rangle_j$. 为了消除恶意委员会成员上传恶意 FVSS 份额的影响, 每个委员会成员 $V_j \in V^*$ 在 $|V^*|$ 个 FVSS 份额中遍历 m 个份额的所有组合, 通过 $FVSS.Recon$ 得到每种组合的全局模型. 根据组合公式, 至少有 C_{m+1}^m 种组合的计算结果是相同的, 即为下一轮的全局模型.

6　安全性分析

本节从隐私性和拜占庭鲁棒性两个方面分析 SDFL 的安全性.

6.1　隐私性分析

SDFL 的隐私性指的是攻击者无法获取客户端的本地更新, 从而防止攻击者从本地更新中推理出客户端的本地数据隐私信息. SDFL 的隐私性依赖于 FVSS 的安全性和 zk-SNARK 的零知识性. 具体地, 在 SDFL 中, 每个客户端 P_i 通过安全信道向每个委员会成员 V_j 发送 $(\langle\theta_i^{(t)}\rangle_j, \pi_{ij})$, 其中 $\langle\theta_i^{(t)}\rangle_j$ 是本地更新 $\theta_i^{(t)}$ 的第 j 个 (m,k)-FVSS 份额, π_{ij} 是以 $\theta_i^{(t)}$ 和 FVSS 多项式系数为证据 (即秘密输入) 生成的零知识证明. 此外, P_i 在公告板上公开 FVSS 多项式系数的承诺 $\Psi_i^{(t)}$.

首先, 根据 (m,k)-FVSS 的安全性 [39], 至少需要 m 个份额才可以计算出 $\theta_i^{(t)}$ 的值, 而在 SDFL 中, 最多存在 $m-1$ 个攻击者. 在最坏情况下, 即在 $m-1$ 个攻击者都是委员会成员的情况下, 即使所有攻击者共谋, 攻击者最多持有 $m-1$ 个 FVSS 份额, 无法得到 $\theta_i^{(t)}$ 的任何信息. 另一方面, 在离散对数困难问题假设下, 攻击者无法通过 FVSS 的承诺 $\Psi_i^{(t)}$ 得到 FVSS 的多项式系数.

其次, 根据 zk-SNARK 的零知识性 [41], 对于任何满足关系 $\mathcal{R}_C$ 的陈述和证据 (x,w), 存在多项式时间模拟器 Sim, 使得模拟器生成的证明 $\pi \leftarrow Sim(trap, x)$ 和真实的证明 $\pi \leftarrow ZK.Prove(PK, x, w)$ 是计算不可区分的, 其中 $trap$ 是模拟器陷门信息. 也就是说, 在 zk-SNARK 中, 攻击者无法从证明 π 得到证据 x 的任何信息. 因此, 具体在 SDFL 中, 任何委员会成员 V_j 都无法从其持有的证明 π_{ij} 中得到客户端 P_i 的本地更新 θ_i 及 FVSS 多项式系数的任何信息.

综合以上分析, 可以得出结论: 在离散对数困难问题假设下及所采用的 zk-SNARK 方案满足零知识性的条件下, 当系统中攻击者数量少于 m 时, SFDL 具有隐私性.

6.2　拜占庭鲁棒性分析

SDFL 的拜占庭鲁棒性指的是系统能防止可能存在的恶意客户端和恶意委员会成员破坏模型聚合结果. 在 SDFL 中, 恶意客户端可能在本地训练阶段执行恶意操作, 而恶意委员会成员可能在模型聚合阶段执行恶意操作.

首先, 针对恶意客户端, SDFL 的拜占庭鲁棒性由 zk-SNARK 的完备性和可靠性保障. 完备性指对于任何的 $(x,w) \in \mathcal{R}_C$, 若 $(PK, VK) \leftarrow ZK.KeyGen(1^\lambda, C)$, $\pi \leftarrow ZK.Prove(PK, x, w)$, 则 $\Pr[1 \leftarrow ZK.Verify(VK, x, \pi)] = 1 - negl(\lambda)$, 其中 $negl(\lambda)$ 是可忽略函数. 也就是说, 只要陈述和证据满足给定的关系, 则生成的证明一定能够验证通过 (严格地说, 验证不通过的概率是可忽略的). 可靠性指对于任何多项式时间攻击者 $\mathcal{A}$ 和提取器 $\mathcal{E}$, 若 $(PK, VK) \leftarrow ZK.KeyGen(1^\lambda, C)$, $(x, \pi) \leftarrow \mathcal{A}(PK, VK)$, $w \leftarrow \mathcal{E}(PK, VK)$, $(x,w) \notin \mathcal{R}_C$, 则 $\Pr[1 \leftarrow ZK.Verify(VK, x, \pi)] = negl(\lambda)$. 如果陈述和证据不满足给

定的关系, 则生成的证明通过验证的概率是可忽略的. 具体地, 在 SDFL 中, 客户端 P_i 向委员会成员 V_j 发送证明 π_{ij}, 根据 zk-SNARK 的完备性和可靠性, 当且仅当 P_i 的本地更新 $\theta_i^{(t)}$ 的 L_2 范数小于设定的阈值 τ, 且 P_i 发送给 V_j 的 FVSS 份额 $\langle\theta_i^{(t)}\rangle_j$ 是从 $\theta_i^{(t)}$ 计算得到时, π_{ij} 才能通过验证.

其次, 针对恶意的委员会成员, SDFL 假设最多存在 $m-1$ 个攻击者, 且 $k \geqslant 2m$, 即委员会中诚实成员占多数. 在恶意客户端检测阶段, 每个委员会成员独立对客户端更新进行验证, 通过信任矩阵达成对合法客户端及合法委员会成员的共识; 在聚合阶段, 通过遍历 m 个份额的组合, 确保聚合结果不被少数恶意委员会成员操纵.

综合以上分析, 可以得出结论: 在所采用的 zk-SNARK 方案满足完备性和可靠性的条件下, 当系统中委员会成员数量大于等于 $2m$ 时, SFDL 具有拜占庭鲁棒性.

7　实验分析

本节对 SDFL 方案进行实验分析, 包括 SDFL 在拜占庭攻击下的分类准确率, 以及系统的计算和通信开销. 实验平台为配置 Intel(R) Xeon(R) Gold 5218R CPU、128 GB 内存、Ubuntu 20.04 操作系统的计算机. 实验使用了 Python 和 C++ 两种编程语言, 采用 Flower Library 框架 [46] 实现联邦学习; 零知识证明采用 Groth 16 方案 [41], 利用 Libsnark Library 框架 (BN 128 曲线) [45] 实现.

7.1　实验设置

7.1.1　数据集和模型结构

为评估所提出的 SDFL 在不同联邦学习模型下的性能表现, 本文选用了两个经典的公开图像数据集: MNIST 和 Fashion-MNIST (FMNIST).

在模型选择方面, 本文分别采用了 3 种典型的机器学习模型进行实验: 逻辑回归 (logistic regression, LR)、多层感知机 (multilayer perceptron, MLP) 和卷积神经网络 (convolutional neural network, CNN). 其中, LR 模型包含一个输入层与一个输出层, 共有 7850 个参数; MLP 模型的输入层为 784 维, 包含一个具有 20 个神经元的隐藏层和一个具有 10 个神经元的输出层, 共有 15910 个参数; CNN 模型由一个 8×8 的卷积层和一个 4×4 的卷积层构成, 每个卷积层后连接 ReLU 激活函数和最大池化层, 随后连接两个全连接层, 分别包含 32 个和 10 个神经元, 共有 26010 个参数. 在模型训练方面, 本文采用小批量随机梯度下降算法, 其中, LR 的批量值为 1024, 学习率随轮次动态变化 [21]; MLP 和 CNN 的批量值为 64, 学习率为 0.01. 为了使用 FVSS 和 zk-SNARK, 在训练过程中把模型参数放大 10^4 倍后取整, 映射为有限域 $\mathbb{Z}_p$ 中的元素; 相应地, LR, MLP 和 CNN 的合法本地更新 L_2 范数边界平方值 (即 τ^2) 分别设置为 20×10^8, 100×10^8 和 200×10^8.

7.1.2　客户端与攻击场景设置

实验默认设置 $n=20$, $k=10$, $m=5$, 即系统中共有 20 个客户端, 其中 10 个客户端组成委员会, 且存在 $m-1=4$ 个攻击者 (20% 攻击者). 为了覆盖所有可能的攻击场景, 假设 4 个攻击者中有 2 个为恶意客户端, 他们在本地训练阶段对系统进行拜占庭攻击, 具体为采用随机生成的高斯噪声替代本地更新 [16,47]; 另外 2 个为恶意委员会成员, 他们在本地训练阶段表现为合法客户端, 会提供正确的本地更新, 但在恶意客户端检测阶段或本地更新聚合阶段对系统进行攻击, 具体为在恶意客户端检测阶段随机构造错误的信任向量或在聚合阶段采用随机生成的高斯噪声替代 FVSS 份额; 在实验中, 除了恶意客户端, 其他客户端随机均分 MNIST 和 FMNIST 的训练样本作为本地训练集.

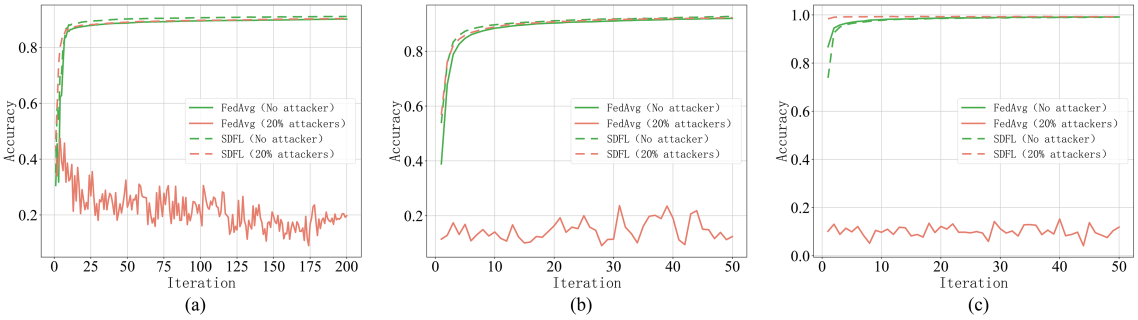

图 4　(网络版彩图) SDFL 在 MNIST 上的分类准确率. (a) LR 模型; (b) MLP 模型; (c) CNN 模型.

Figure 4　(Color online) Classification accuracy of SDFL on MNIST. (a) LR model; (b) MLP model; (c) CNN model.

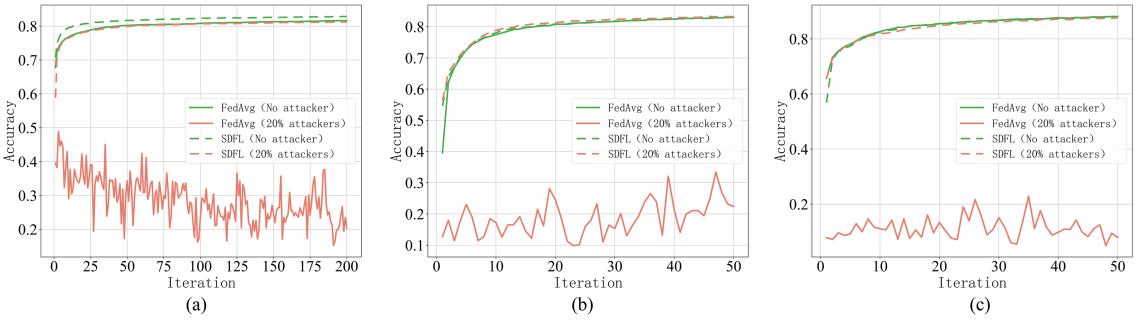

图 5　(网络版彩图) SDFL 在 FMNIST 上的分类准确率. (a) LR 模型; (b) MLP 模型; (c) CNN 模型.

Figure 5　(Color online) Classification accuracy of SDFL on FMNIST. (a) LR model; (b) MLP model; (c) CNN model.

7.2　分类准确率

图 4 和 5 分别展示了 SDFL 在 MNIST 与 FMNIST 两个数据集上的分类性能. 针对每个数据集, 分别采用 LR, MLP 和 CNN 3 种模型进行测试, 并将结果与经典联邦学习算法 FedAvg [1] 进行对比. 实验结果表明, 在设定的攻击场景下, FedAvg 的分类准确率最高不超过 34%, 表现明显下降, 而 SDFL 在存在 20% 攻击者的情况下, 其分类准确率与无攻击者场景下基本一致, 且接近 FedAvg 在无攻击者条件下的表现, 显示出良好的抗拜占庭攻击能力. 例如, 在 CNN 和 MNIST 的实验中, SDFL 在攻击者比例为 20% 时的分类准确率仍达到 99%. 此外, 实验结果还验证了 SDFL 对多种模型结构的兼容能力, 证明其具有良好的通用性与可扩展性.

图 6 进一步展示了 SDFL 与最新的基于差分隐私的去中心化联邦学习方案 [21] 的分类准确率对比结果. 由于文献 [21] 假设损失函数是凸的, 在该实验中仅对比 LR 模型的分类准确率. 具体地, 该实验采用了与文献 [21] 相同的设置, 系统中共有 12 个客户端, 其中 2 个客户端是恶意的, 在本地训练中上传高斯噪声替代本地更新. 文献 [21] 支持多种拜占庭检测方法, 本文选择其中分类准确率最高的 IOS 方法 [48] 进行对比 (图 6 中的 IOS+noise). 实验结果表明, 在相同的拜占庭攻击场景下, SDFL 的分类准确率明显高于基于差分隐私的方案. 例如, 在 MNIST 数据集上, SDFL 的分类准确率超过 90%, 而文献 [21] 的分类准确率约为 85%.

SDFL 假设客户端数量 (n)、委员会成员数量 (k) 和攻击者数量 ($m-1$) 满足 $n \geqslant k \geqslant 2m$. 为了进一步验证攻击者数量对模型分类性能的影响, 我们对不同攻击者数量下的模型分类准确率进行了测试, 实验结果如图 7 和 8 所示. 由于恶意客户端的数量不会对恶意客户端检测和本地更新聚合产生影响, 我们假设所有攻击者都是恶意委员会成员. 实验结果表明, 当攻击者数量满足假设条件时, SDFL 在 MNIST 和 FMINST 数据集上的分类准确率与无攻击者时相同; 而当攻击者数量超过假设条件时 (即图 7 和 8 中攻击者数量等于 10 时), 3 种模型在两个数据集上的分类准确率都明显下降, 因为此

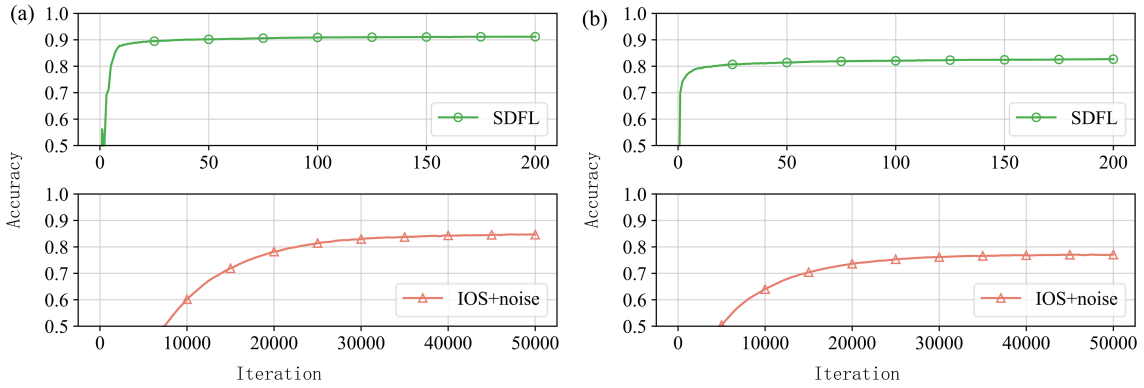

图 6　(网络版彩图) SDFL 与基于 DP 的方案 [21] 的分类准确率对比. (a) LR 模型, MNIST 数据集; (b) LR 模型, FMNIST 数据集.

Figure 6　(Color online) Comparison of classification accuracy between SDFL and DP-based schemes [21]. (a) LR model, MNIST dataset; (b) LR model, FMNIST dataset.

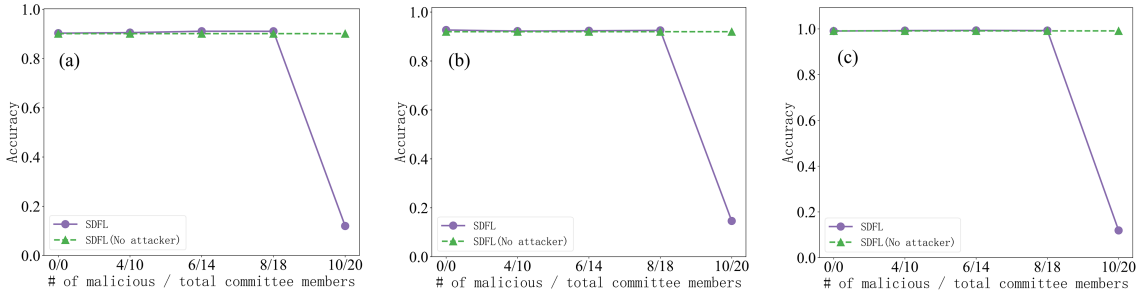

图 7　(网络版彩图) 不同攻击者数量对 SDFL 分类准确率的影响 (MNIST). (a) LR 模型; (b) MLP 模型; (c) CNN 模型.

Figure 7　(Color online) Impact of the number of attackers on the classification accuracy of SDFL (MNIST). (a) LR model; (b) MLP model; (c) CNN model.

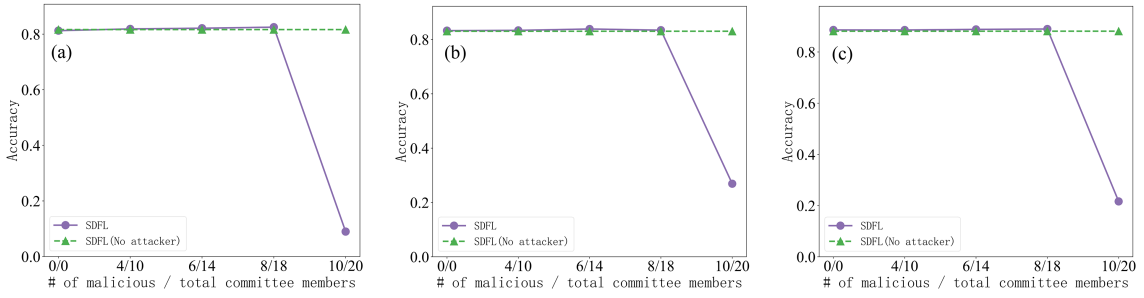

图 8　(网络版彩图) 不同攻击者数量对 SDFL 分类准确率的影响 (FMNIST). (a) LR 模型; (b) MLP 模型; (c) CNN 模型.

Figure 8　(Color online) Impact of the number of attackers on the classification accuracy of SDFL (FMNIST). (a) LR model; (b) MLP model; (c) CNN model.

时合法委员会成员无法在本地更新聚合阶段通过不同的 FVSS 份额组合得到正确的聚合结果. 因此, SDFL 适用于合法委员会成员占大多数的情况.

7.3　计算与通信开销

除了分类性能分析, 本文对 SDFL 的计算与通信开销也进行了详细的测试. 由于 MNIST 和 FMINST 两个数据集的样本数量和样本维度相同, 所以该部分实验仅采用 MNIST 数据集.

表 2　SDFL 的计算开销 (单位: 秒).

Table 2　Computation overhead of SDFL (in seconds).

Model	Param.	Local training			Aggregation		
		ClientUpdate	FVSS.Share	ZK.Prove	FVSS.Verify	ZK.Verify	FVSS.Recon
LR	7850	1.1	11.9	4.2	7.5	0.2	5.4
MLP	15910	9.8	24.2	9.1	15.2	0.4	10.9
CNN	26010	30.7	40.8	13.3	25.1	0.5	17.0

表 3　SDFL 的客户端通信开销.

Table 3　Communication overhead of the client in SDFL.

Model	Param.	FVSS share (MB)	FVSS commitment (MB)	ZKP proof (KB)
LR	7850	0.44	12.1	0.13
MLP	15910	0.89	24.6	0.13
CNN	26010	1.50	40.2	0.13

7.3.1　计算开销

在 SDFL 中, 除模型本地更新外, 系统的主要计算开销来自方案中引入的密码学操作, 具体包括本地训练阶段的 FVSS 份额及承诺生成、零知识证明生成, 模型聚合阶段的 FVSS 份额验证、零知识证明验证, 以及不同组合的 FVSS 秘密恢复 (即聚合模型的恢复). 如表 2 所示, 在 SDFL 中, 以上各项密码学操作的计算开销随着模型参数量的增加呈现线性增长趋势. 其中, FVSS 份额与承诺生成及 FVSS 份额验证的时间开销最为显著, 原因在于 FVSS 的承诺生成及份额验证操作涉及大量的模指数运算. 以 CNN 为例, 明文本地更新的计算耗时为 30.7 s, 而各项密码学操作共耗时 96.7 s, 计算开销增长约 3 倍, 表明 SDFL 在保障联邦学习安全与隐私的同时保持了较好的计算性能. 然而, 也应注意, 对于 LR 这类参数量较少的模型, SDFL 相较于明文操作的计算开销增加相对较大.

7.3.2　通信开销

除了计算开销, 本文还对 SDFL 中客户端的主要通信开销进行了评估. SDFL 中客户端的通信开销主要包括三部分: 发送给委员会成员的本地更新的 FVSS 份额与零知识证明, 以及发送到公告板的 FVSS 承诺. 具体实验结果如表 3 所示, 可以看出, FVSS 份额和承诺的大小随着模型参数量的增加而线性增加; 而得益于 Groth 16 零知识证明方案的特性, 零知识证明的大小保持不变. 总体而言, SDFL 客户端的通信开销相对较小, 在当前网络带宽条件下通信延迟基本可以忽略. 需要说明的是, 在 SDFL 中, 除了客户端的通信开销外, 委员会成员还会将聚合结果的 FVSS 份额上传到公告板, 以及从公告板下载其他委员会成员上传的 FVSS 份额, 但这部分的单个 FVSS 份额大小与客户端发送给委员会成员的份额大小相同, 因此不再单独列出.

以上实验结果表明, SDFL 在引入较小计算与通信开销的前提下, 能够有效抵抗拜占庭攻击, 具有良好的拜占庭鲁棒性和高效性.

8　结论

本文针对去中心化联邦学习中面临的隐私泄露与拜占庭攻击风险, 提出了一种同时具备隐私保护与抗拜占庭攻击能力的去中心化联邦学习方案 SDFL. 该方案在客户端 – 委员会架构下, 结合 FVSS 与零知识证明技术, 实现了无需中央服务器的本地更新隐私保护与恶意行为检测. 理论和实验分析表明, SDFL 在不损失模型分类准确率的前提下, 具有隐私性、拜占庭鲁棒性和计算高效性, 有效提升了

去中心化联邦学习的安全性. SDFL 采用 L_2 范数边界作为拜占庭检测方法, 在未来工作中, 我们将探索采用防御能力更强的检测方法, 并进一步提高系统的高效性.

参考文献

1 McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 2017. 1273–1282

2 Gu Y H, Bai Y B. Survey on security and privacy of federated learning models. J Softw, 2023, 34: 2833–2864 [顾育豪, 白跃彬. 联邦学习模型安全与隐私研究进展. 软件学报, 2023, 34: 2833–2864]

3 Gao Y, Chen X F, Zhang Y Y, et al. A survey of attack and defense techniques for federated learning systems. Chin J Comput, 2023, 45: 1781–1805 [高莹, 陈晓峰, 张一余, 等. 联邦学习系统攻击与防御技术研究综述. 计算机学报, 2023, 46: 1781–1805]

4 Bai L, Hu H, Ye Q, et al. Membership inference attacks and defenses in federated learning: a survey. ACM Comput Surv, 2024, 57: 1–35

5 Zhao J C, Sharma A, Elkordy A R, et al. Loki: large-scale data reconstruction attack against federated learning through model manipulation. In: Proceedings of the 2024 IEEE Symposium on Security and Privacy (SP), 2024. 1287–1305

6 Li Y, Yan H Y, Huang T, et al. Model architecture level privacy leakage in neural networks. Sci China Inf Sci, 2024, 67: 132101

7 Wang B, Dai X R, Wang W, et al. Adversarial examples for poisoning attacks against federated learning. Sci Sin Inform, 2023, 53: 470–484 [王波, 代晓蕊, 王伟, 等. 面向联邦学习的对抗样本投毒攻击. 中国科学: 信息科学, 2023, 53: 470–484]

8 Gong X, Chen Y, Wang Q, et al. Backdoor attacks and defenses in federated learning: state-of-the-art, taxonomy, and future directions. IEEE Wireless Commun, 2023, 30: 114–121

9 Beltrán E T M, Pérez M Q, Sánchez P M S, et al. Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. IEEE Commun Surv Tutor, 2023, 25: 2983–3013

10 Mansouri M, Önen M, Jaballah W B, et al. Sok: secure aggregation based on cryptographic schemes for federated learning. In: Proceedings on Privacy Enhancing Technologies, 2023

11 Xie Q, Jiang S, Jiang L, et al. Efficiency optimization techniques in privacy-preserving federated learning with homomorphic encryption: a brief survey. IEEE Internet Things J, 2024, 11: 24569–24580

12 Fang M, Zhang Z, Hairi, et al. Byzantine-robust decentralized federated learning. In: Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security, 2024. 2874–2888

13 Tang P, Zhu X, Qiu W, et al. FLAD: Byzantine-robust federated learning based on gradient feature anomaly detection. IEEE Trans Dependable Secure Comput, 2025, 22: 3993–4009

14 Miao Y B, Kuang D, Li X H, et al. Efficient privacy-preserving federated learning under dishonest-majority setting. Sci China Inf Sci, 2024, 67: 159102

15 Mu X T, Cheng K, Song A X, et al. Privacy-preserving federated learning resistant to byzantine attacks. Chin J Comput, 2024, 47: 842–861 [穆旭彤, 程珂, 宋安霄, 等. 抗拜占庭攻击的隐私保护联邦学习. 计算机学报, 2024, 47: 842–861]

16 So J, Guler B, Avestimehr A S. Byzantine-resilient secure federated learning. IEEE J Sel Areas Commun, 2020, 39: 2168–2181

17 Lycklama H, Burkhalter L, Viand A, et al. RoFL: robustness of secure federated learning. In: Proceedings of the 2023 IEEE Symposium on Security and Privacy (SP), 2023. 453–476

18 Roy Chowdhury A, Guo C, Jha S, et al. EIFFeL: ensuring integrity for federated learning. In: Proceedings of 2022 ACM SIGSAC Conference on Computer and Communications Security, 2022. 2535–2549

19 Gu X, Li M, Xiong L. DP-BREM: differentially-private and byzantine-robust federated learning with client momentum. In: Proceedings of the 34th USENIX Security Symposium (USENIX Security 25), 2025

20 Qi T, Wang H, Huang Y. Towards the robustness of differentially private federated learning. In: Proceedings of the 38th AAAI Conference on Artificial Intelligence, 2024

21 Ye H, Zhu H, Ling Q. On the tradeoff between privacy preservation and byzantine-robustness in decentralized learning. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2024. 9336–9340

22 Zhang A, Zhao P, Lu W, et al. Personalized decentralized federated learning: a privacy-enhanced and byzantine-resilient approach. IEEE Trans Comput Soc Syst, 2025, 12: 3206–3217

23 Zhou S, Huang H, Li R, et al. ComAvg: robust decentralized federated learning with random committees. Comput Commun, 2023, 211: 147–156

24 Che C, Li X, Chen C, et al. A decentralized federated learning framework via committee mechanism with convergence guarantee. IEEE Trans Parallel Distrib Syst, 2022, 33: 4783–4800

25 Guo J, Liu Z, Tian S, et al. TFL-DT: a trust evaluation scheme for federated learning in digital twin for mobile networks. IEEE J Sel Areas Commun, 2023, 41: 3548–3560

26 Yang L, Miao Y, Liu Z, et al. Enhanced model poisoning attack and multi-strategy defense in federated learning. IEEE Trans Inform Forensic Secur, 2025, 20: 3877–3892

27 Huo W, Yu Y, Yang K, et al. Privacy-preserving cryptographic algorithms and protocols: a survey on designs and applications. Sci Sin Inform, 2023, 53: 1688–1733 [霍炜, 郁昱, 杨糠, 等. 隐私保护计算密码技术研究进展与应用. 中国科学: 信息科学, 2023, 53: 1688–1733]

28 Fu J, Hong Y, Ling X, et al. Differentially private federated learning: a systematic review. ArXiv:2405.08299

29 Bünz B, Bootle J, Boneh D, et al. Bulletproofs: short proofs for confidential transactions and more. In: Proceedings of the 2018 IEEE symposium on security and privacy (SP), 2018. 315–334

30 Bonawitz K, Ivanov V, Kreuter B, et al. Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017

31 Corrigan-Gibbs H, Boneh D. Prio: private, robust, and scalable computation of aggregate statistics. In: Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17), 2017. 259–282

32 Xie C, Koyejo S, Gupta I. Zeno++: robust fully asynchronous SGD. In: Proceedings of the 37th International Conference on Machine Learning, 2020. 10495–10503

33 Wang F, He Y, Guo Y, et al. Privacy-preserving robust federated learning with distributed differential privacy. In: Proceedings of the 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2022. 598–605

34 Zhu H, Ling Q. Bridging differential privacy and byzantine-robustness via model aggregation. In: Proceedings of the 31st International Joint Conference on Artificial Intelligence, 2022. 2427–2433

35 Li L, Xu W, Chen T, et al. RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2019. 33: 1544–1551

36 Karimireddy S P, He L, Jaggi M. Learning from history for byzantine robust optimization. In: Proceedings of the 38th International Conference on Machine Learning, 2021. 5311–5319

37 Naseri M, Hayes J, de Cristofaro E. Local and central differential privacy for robustness and privacy in federated learning. In: Proceedings of the Network and Distributed Systems Security (NDSS) Symposium, 2022

38 Xie C, Long Y, Chen P Y, et al. Unraveling the connections between privacy and certified robustness in federated learning against poisoning attacks. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, 2023. 1511–1525

39 Feldman P. A practical scheme for non-interactive verifiable secret sharing. In: Proceedings of the 28th Annual Symposium on Foundations of Computer Science (SFCS 1987), 1987. 427–438

40 Goldwasser S, Micali S, Rackoff C. The knowledge complexity of interactive proof-systems. In: Proceedings of the Providing Sound Foundations for Cryptography: on the Work of Shafi Goldwasser and Silvio Micali, 2019. 203–225

41 Groth J. On the size of pairing-based non-interactive arguments. In: Proceedings of the EUROCRYPT 2016, 2016. 305–326

42 Sabater C, Hahn F, Peter A, et al. Private sampling with identifiable cheaters. In: Proceedings on Privacy Enhancing Technologies, 2023

43 Sun Z, Kairouz P, Suresh A T, et al. Can you really backdoor federated learning? ArXiv:1911.07963

44 Rathee M, Shen C, Wagh S, et al. ELSA: secure aggregation for federated learning with malicious actors. In: Proceedings of the 2023 IEEE Symposium on Security and Privacy (SP), 2023. 1961–1979

45 SCIPR. Libsnark: a C++ library for zkSNARK proofs. https://github.com/scipr-lab/libsnark

46 Beutel D J, Topal T, Mathur A, et al. Flower: a friendly federated learning research framework. ArXiv:2007.14390

47 Blanchard P, El Mhamdi E M, Guerraoui R, et al. Machine learning with adversaries: Byzantine tolerant gradient descent. In: Proceedings of Advances in Neural Information Processing Systems 30 (NIPS 2017), 2017. 30

48 Wu Z, Chen T, Ling Q. Byzantine-resilient decentralized stochastic optimization with robust aggregation rules. IEEE Trans Signal Process, 2023, 71: 3179–3195

SDFL: a privacy-preserving Byzantine-robust decentralized federated learning scheme

Hanyu QUAN[1,2], Yanyi QIAN[1,2], Hui TIAN[1,2*], Xuefeng LIU[3], Yue LI[1,2], Jianzong WANG[4] & Jia ZHONG[5]

1. *College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China*
2. *Xiamen Key Laboratory of Data Security and Blockchain Technology, Xiamen 361021, China*
3. *School of Cyber Engineering, Xidian University, Xi'an 710071, China*
4. *Ping An Technology (Shenzhen) Co., Ltd., Shenzhen 518046, China*
5. *Quangong Machinery Co., Ltd., Quanzhou 362123, China*

* Corresponding author. E-mail: htian@hqu.edu.cn

Abstract Decentralized federated learning (DFL) eliminates reliance on a central server, thereby addressing the single point of failure inherent in traditional federated learning frameworks. However, this decentralization also amplifies the risks of privacy leakage and Byzantine attacks. Existing research on privacy protection and Byzantine robustness has primarily focused on centralized federated learning and cannot be directly applied to decentralized settings. To address this gap, we propose SDFL, a privacy-preserving and Byzantine-robust decentralized federated learning scheme. SDFL adopts a client–committee architecture, where model aggregation is performed by randomly selected committees. We design a privacy-preserving and Byzantine-robust method based on verifiable secret sharing and the zero-knowledge proof technique. Security analysis demonstrates that SDFL provides strong guarantees of both privacy and robustness in decentralized environments. Experimental evaluations on the MNIST and fashion-MNIST datasets with multiple machine learning models show that SDFL effectively mitigates Byzantine behaviors while maintaining high classification accuracy and efficiency.

Keywords federated learning, privacy protection, Byzantine robustness, verifiable secret sharing, zero-knowledge proof